Draft 1,
24 Feb. 2004

IFC Model Server Technology

Francois Grobler,
ERDC-Champaign

Introduction

This paper examines open, model-based, web servers to serve Facility Management and Facility Area Networks. The IFC (Industry Foundation Classes) specification of the International Alliance for Interoperability (IAI) is briefly introduced and arguments are presented for using a common object model in the evolving Building Information Model (BIM) and FAN (Facility Area Networks). The status of IFC model servers is reviewed and conclusions are presented on existing and developing capabilities.

Background

The International Alliance for Interoperability (IAI) has been actively pursuing interoperability in the Architecture, Engineering, Construction and Facility Management industries for 8 years. The IAI is a non-profit global alliance of the building, research and information technology industries working to enable and promote interoperability (sharing information via integrated technological solutions) across disciplines, technical applications and the facility life cycle.
The IAI is dedicated to improving processes, enabling integrated information exchange and encouraging collaborative working within the industry. Organizations within the alliance include architecture, engineering, construction (AEC), building owners and maintainers, facility managers (FM), product manufacturers, software vendors, information providers, government agencies, trade and professional associations, research laboratories, and universities.

A major part of the IAI's mission is to define, promote and publish the IFC (Industry Foundation Classes) Model specification, which enables IT companies to integrate interoperable functionalities into their software applications, tools and systems. The IFCs are based on non-proprietary, freely available, international open standards, have regular releases, and provide a global software certification process. The IAI continues to work on additional and improved functionalities to be included in upcoming IFC releases. The IAI also develops XML schema (e.g., ifcXML or aecXML) to complement the IFC Model, enhance Internet communication and e-commerce transactions.

What does interoperability mean?

Interoperability is the sharing and exchanging of information via integrated technological solutions, no matter what project phase, discipline or participant role in the building life cycle. It involves innovative and integrated concepts of real time, on-line 3D/4D virtual projects, comprehensive project data, value adding capabilities, project partner collaboration (clients, service providers, product suppliers, end users) and stretches across technical applications, tools and systems. The object oriented, intelligent data allows attributes such as geometry, relationships, code compliance, materials, cost and time factors to be added and maintained within the virtual model. It integrates with web-

based technologies and allows each participant to use technologies and software specific to their needs without losing, compromising or corrupting project data.
The benefits of interoperability in the building community include improved productivity, competitive advantage, increased profitability, integrated processes, seamless information flow, whole-of-project approach and enhanced capabilities. It also incorporates improved, collaborative processes that result in reduced time, cost and rework by all project participants throughout all phases of the life of the facility.

Data exchange in IFC

Interoperability is enabled by the IFC standard that defines the common semantics for a shared data model used by project participants. In order to exchange information freely the messages (data packets) must be based on this shared data model. However, in IFC several syntaxes are acceptable. In file-based exchanges between IFC compatible software the data is exchanged as in "STEP Physical File" (SPF) format as defined by ISO1030-Part 21). The IFC model is also expressed in XML, i.e. ifcXML is automatically generated from the IFC model (documented in the EXPRESS language per ISO 10303-Part 11) using a methodology and software developed by the IAI, based on an ISO translation methodology for generating XML from EXPRESS product models (ISO 10303-Part 28). ifcXML is extensively used by the model servers described below for Internet based communication of data. aecXML is an effort of the North American IAI to develop schemas for North American needs based on ifcXML, while incorporating elements of XML already in use for commerce purposes (i.e. XML that does not overlap in scope with ifcXML).

Building Information Model

Recently the concept of Building Information Model (BIM) was popularized in the trade press, but the concept has a much longer lineage. The notion to store all project relevant data over its lifecycle can be traced to at least the NSF sponsored Woods Hole Workshops in the 1970's and perhaps earlier. The IAI community has been pursuing the same concept since the beginning.

Many of the benefits mentioned above can only be realized in the context of a shared data storage capability that covers the facility life-cycle and allow access to relevant data to participants in the process as needed, while offering the owner of the facility the control needed to:

1) Safeguard this valuable information.
2) Archive this information in a vendor independent format (a big issue with government owner who routinely own facilities for decades, or even centuries)
3) Exchange software conveniently when newer, more powerful software becomes available.

Figure 1 depicts the concepts of a BIM in the center of the facility life cycle, providing data access to the many business processes of interest along the way.

While a firm definition for BIM has not emerged yet, this paper proposes the following definition:

A computable representation of the physical and functional characteristics of a facility, and its related project/life-cycle information. It integrates all the relevant aspects into a coherent organization of data that computer applications can access, modify and/or add to it, if authorized to do so, using open industry standards.

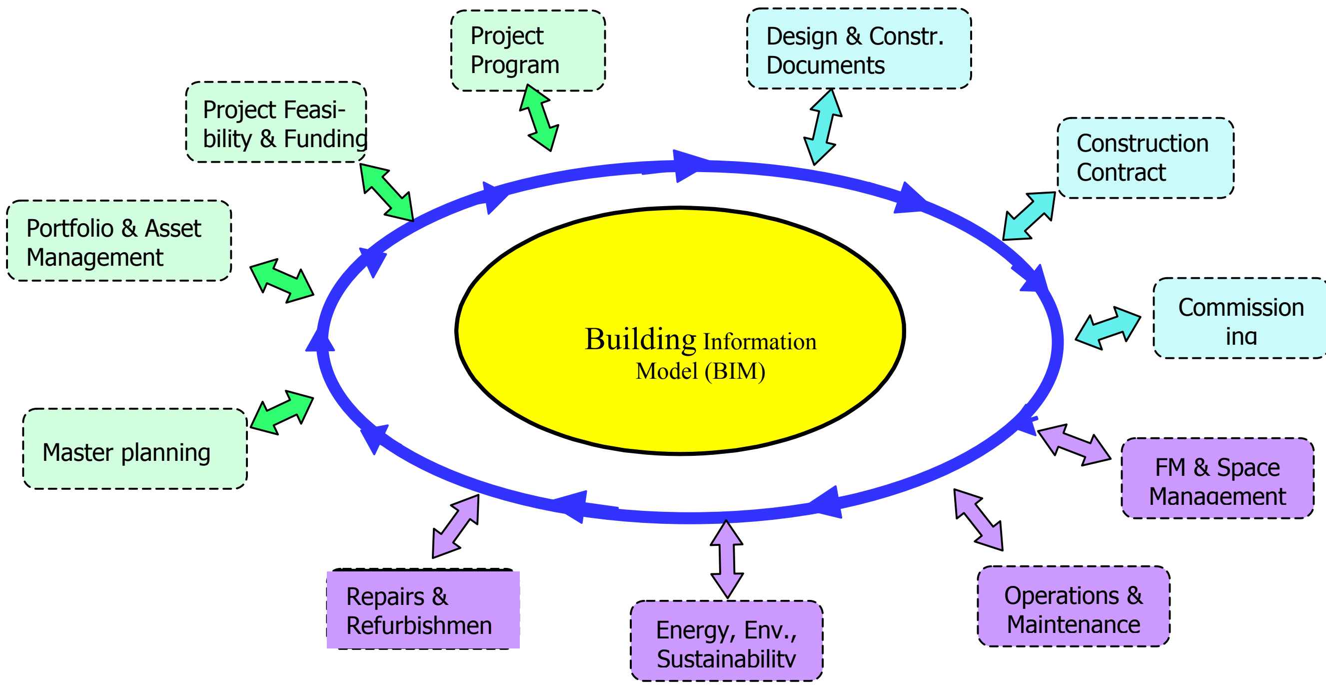

Figure 1: Building Information Model

Why a model server?

A large number of software products have implemented support for file-based exchanges of IFC model project data, i.e. IFC read and write functionality. These exchanges have clearly demonstrated the benefits of using common semantics and syntax in exchanging common information about projects, and pilot projects have shown that the file exchange does serve certain situations well. However, when file based exchange is used in real projects the limitations of file exchange become evident very quickly. If there are a large number of participants, such as real project invariable have, the management required to make sure each participant is using the right version of the files becomes impractical. Additionally, managing access rights, versioning and the right to make and save changes become a significant burden, is cumbersome and prone to human mistakes.

Additionally, direct communication over the Web, from application to application is not feasible.

What is a model server?

The Web-based model server approach solves these problems. The model server provides a central repository of the shared information in the IFC model and all authorized users have instant access to the most up-to-date information that is relevant to their disciple from anywhere where Web services are available, using any device. (IFC model server access has been demonstrated via mobile phone in Japan to a server in Finland). Additionally data management features such as partial model exchange, merge, append and changes tracking become possible. It is these additional features that make model servers much more than a document management system.

Advantages of using a model server

The advantages of using a model sever in conjunction with FAN are important and pervasive. While it is possible to provide significant enhancements to various operations and transactions without using the common semantics of an underlying object model, the real power of interoperability is only realized when that is the case. It is estimated that every piece of information used in the life cycle of a construction project is entered more than seven times, with the commensurate share of errors and distortions. Using different data models for different parts of the process only perpetuates the problem. Object server technology based on IFC models make available the most accurate, up-to-date-information throughout the building life cycle, as the information evolves. Using standard Web services any device that can connect to the Internet, including devices of interest to FAN, can access this information.

One strong benefit of using a BIM on a model server is difficult to quantify – that is the expectation that once all the BIM data is readily available, correct and up-to-date, ingenuity will lead to many new methods of management, analysis and prediction that the industry have not dreamed of yet, and some of the intelligent agent capabilities that we have dreamed about.

Software developers can readily use IFC model services in their software products by creating clients that communicate with the server using standard Web technologies (XML, Web services/SOAP). End users who are mostly interested in "pushing the 'save as' button, don't interact with, and can be unaware of, the server function; for them it simply appears as the ability of the software they are using to share IFC models over the internet.

As a result the proposed FAN element software could be built upon the shared BIM, accessible in an open standards and being developed and populated as the facility moves along its life cycle. The FAN modules could access this information using standard web services with context aware hardware, enabled by the extensive geometry, space, product and other information that is served by the IFC servers.

Model server development in IFC community

The IAI started its efforts towards interoperability with the development of the IFC model, because that type of international consensus standard making can be time consuming. However, it was anticipated from the beginning that model servers will be used in “real world” application and deployment.

A number of middle-ware projects were launched quite early and as a result several product have been developed. For example, the Finnish company Olof Granlund developed “BS Pro” (BS for building services) around 1998 to serve as object server between IFC model based building information and energy simulation software. This product served its intended function rather well and today the DOE and Lawrence Berkeley National Laboratory are still using it for their premier energy simulation engine “EnergyPlus”. BS pro is more aptly described as middleware, not a full-fledged model server.

The second generation of IFC servers was started by the Secom IFC server, developed by the Japanese firm with that name. The purpose of this server was to serve IFC data in a legacy format for use by domestic Japanese software. Later the developer, Yoshinobu Adachi, joined forces with VTT in Finland during a sabbatical and developed IMSvr during 2001 and 2002 [1]. Work on IMSvr was completed in September 2002.

Some of the technology challenges addressed in this project were:

- Storing IFC model data in a database system. (SQL Server, Oracle, etc)
- Automatic schema conversion from EXPRESS to database schema by utilizing XML technology
- XML based SOAP communication between model server and client software

The IMSvr is in the public domain and available for use. Since it is freeware extensive support is not available, but Mr. Adachi has been willing to provide help as needed. A brief review of IMSvr by the Technical Coordinator for the BLIS software development group) is available [2]

Several other commercial efforts started around the same time and resulted in commercial and fully supported products. The most notable products are the EuroStep Model Server for IFC (EMS) [3] and EPM EXPRESS Data Manager (EDM) [4].

EMS is a pure Java application and supports MySQL, SQL Serve and ORACLE databases. The native data access format for EMS is XML per ISO 10303-Part 28. The data interface can be used with http-request, XML-exchanges or with SOAP wrapped massages. It provides a strong set of model server capabilities such as version and change management, document management and linking, model browsing and more.

EDM is a suite of software offering a native object database with interface building and application building utilities. It also offers a basic set of model server type capabilities and access control.

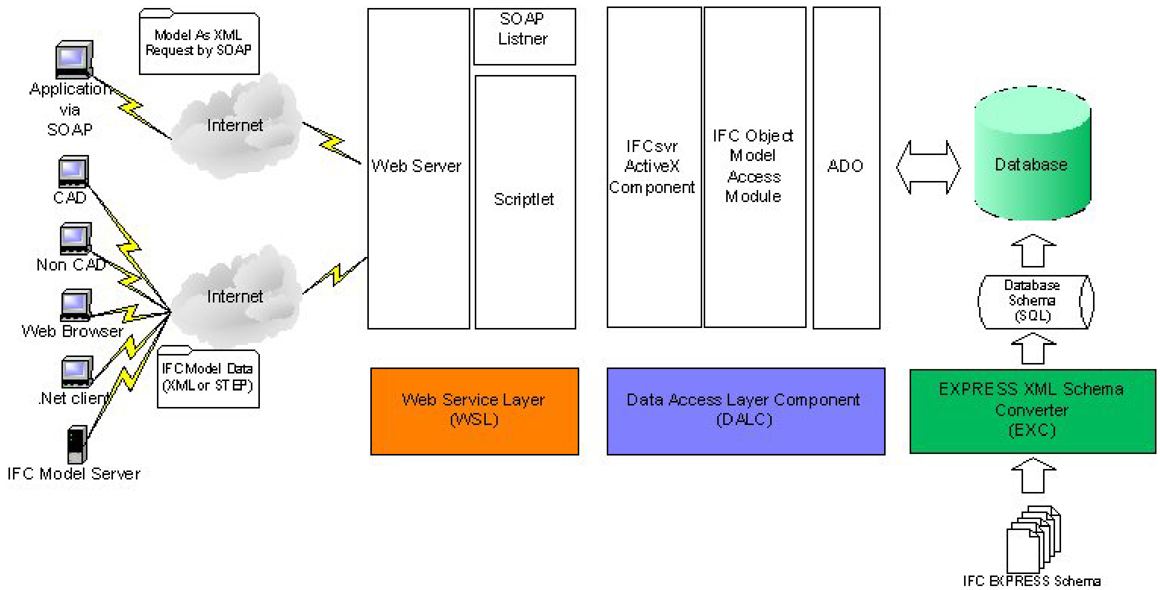

Figure 2: The IMSvr Framework

SABLE project

The SABLE acronym stands for "**S**imple **A**ccess to the **B**uilding **L**ifecycle **E**xchange". The Sable project is a collaborative, open effort to develop model server specifications. Close to 20 companies are participating in the project that started in mid-2003 and is scheduled to complete by mid 2005. According to the SABLE website [5] the objective is:

> *In short SABLE is about harmonizing the access to IFC model servers and the access to the data that persists on these servers by defining a common low level API to IFC Model Servers and a set of high-level domain specifics API to the IFC data model.*

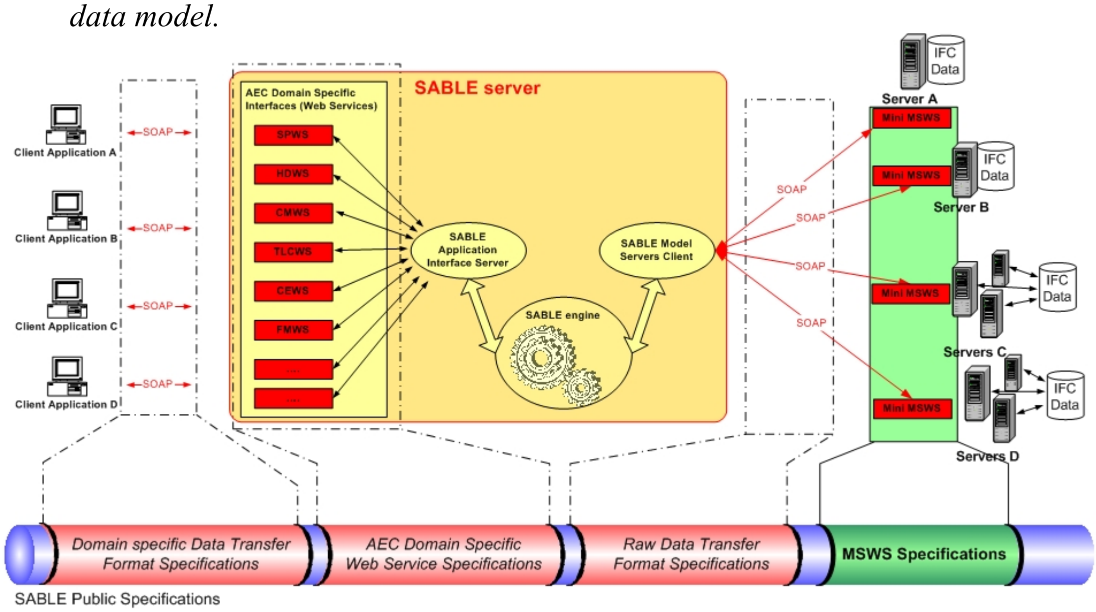

Figure 3: The SABLE Server Concept

SABLE will have the following model server capabilities:

1) Administration – Create and manage projects/facilities; create and manage roles; create and manage participants/users; manage roles in projects and assignment of participant to roles.
2) Data fetch – Get containment tree; object by ID; object by type.
3) Data modification – Create and manage objects and BIM; ownership and access to objects.
4) Documentation – Create, link and get documentation to relevant elements of model and objects.
5) Import/Export – Import and export model data in STEP Physical File (10303, part 21) and XML (10303, part 28 – which is still in Committee Draft status) [6].
6) Information – Set and get project and model information; set and get project and model metadata; get model statistics.
7) Create store and execute partial model queries.
8) Session management – login; select project; model; logout.
9) Versioning – Set and get model versions; manage versions; merge.

In addition to strengthening the expected model server capabilities such as versioning, access control, tracking ownership of objects and change rights, SABLE extends the notion of IFC model server in mainly two ways:

1) SABLE enables a BIM across multiple distributed model servers
2) SABLE facilitates easier IFC compatible software implementation by defining a series of Domain Specific Web Services. The current goals are:
   - Architecture Design Web Service (ADWS)
   - Space Planning Web Service (SPWS)
   - HVAC Design Web Service (HDWS)
   - Construction Management Web Service (CMWS)
   - Thermal Load Calculation Web Service (TLCWS)
   - Quantity take-off Web Service (QTOWS)
   - Cost Estimation Web Service (CEWS)
   - Facility Management Web Service (FMWS)

For the purpose of developing FAN services and BIM use by Government facility owners, it would be important to participate in the business case development and requirements definition of Facility Management Web Services in the SABLE project.

Status of IFC models servers

In summary, currently there are several 2nd generation IFC model servers available from commercial vendors, such as EDM and EMS, as well as IMSvr available in the public domain. These servers store a BIM on a single database server using commercial DBMS such as MySQL, SQL Serve and ORACLE.

The SABLE project is a combined effort of all the players in the category above as well as many other software vendors and user organization, to create a robust, industrial-strength, distributed environment for storing BIM with all the services necessary to

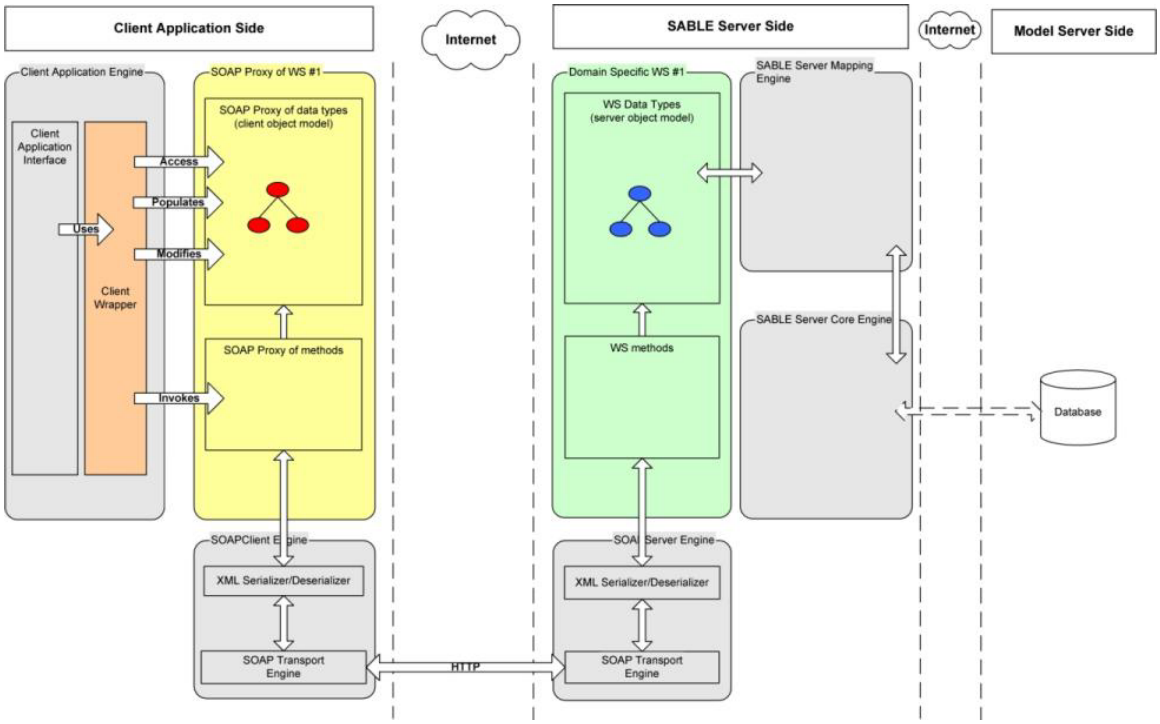

Figure 4: Overview of the SABLE Architecture

facilitate rapid and robust application development. Progress on the ABLE project is according to schedule and expected to conclude by mid-2005. Many of the vendors involved in the project are implementing the SABLE capabilities as the specification are being proposed and proofed, so it is expected that commercial offerings based on the SABLE specifications will become commercially available by late 2005.

Some of the accomplishments of the SABLE project to date are:

- Model Server Web Services (MSWS) Specification published – December 2003
- SABLE High level Resources API "Geometry", "Architecture", "Thermal Analysis" all started in June-July 2003.
- SABLE High level Domain API "Quantity Information for Cost Estimation" started in January 2004.

Conclusions

In order to elevate the AEC+FM industries to the next plateau of efficiency (in fact, to turn around a alarming decline in productivity) the fragmentation of the industry has to be addressed. The most promising approach to this achievement is data sharing, or interoperability, among project participants along the life cycle of facilities.

Facility Area Networks offer new levels of work productivity and quality in the Facilities Ownership, Operation and Maintenance area. However, to maximize these benefits it is important that FANs do not develop new "islands" of interoperability, outside the industry effort by the IAI. It is crucial that FANs utilize the existing and evolving IFC model as the basis for modeling its own data, so that merging it into the larger stream of BIM life cycle information can be accomplished efficiently and accurately.

The IFC model servers that are currently available provide the initial capabilities necessary for developing FAN demonstrations within the context of IFC-based BIM. Emerging capabilities through he SABLE project are likely to fully satisfy the requirements of "real world", deployed FANs, if the requirements are considered in the SABLE project.

The Collaboration with the SABLE project from the FAN development community is recommended to ensure any specific requirements are considered in the specification. The development of the Facility Management Web Service (FMWS) module has not yet started (at this writing in Feb 2005) and this forum would offer the ideal opportunity to elevate any FAN specific requirements for inclusion in the FMWS specification.

References

[1] IMSvr webpage: http://cic.vtt.fi/projects/ifcsvr/
[2] BLIS Review: IMSvr, Jiri Hietanan, http://www.blis-project.org/~sable/ (see Resources and Reviews)
[3] http://www.eurostep.com/
[4] http://www.epmtech.jotne.com
[5] http://www.blis-project.org/~sable/
[6] ISO/CD 10303-28: "Product data representation and exchange: Implementation methods: XML Schema governed representation of EXPRESS schema governed data", June 2003